IE 506: Machine Learning - Principles and Techniques **Jan-May 2024**

Project Mid-term Report : Locally Weighted Ensemble Algorithm

Team Name: Mayur_B *Team Members: 23M1027*

Abstract

The project is about implementing and experimenting with LOCALLY WEIGHTED ENSEMBLE (LWE) ALGORITHM, there are many Ensemble Algorithms such as Bagging, Boosting etc. but they are Globally Weighted Ensembles, which give equal weight to all the base models. In the Locally Weighted Ensemble algorithm we give different weights to different models, based on their ability to predict the data points belonging to different classes. The LWE algorithm has been implemented on the 20 Newsgroup dataset in order to test the performance of the model and compare its performance with the individual base models, SMA (Simple Model Averaging) Framework and LS-SVM (Least Square Support Vector Machine) model.

1 Introduction

The Locally Weighted Ensemble (LWE) is a novel algorithm that combines multiple models to make predictions. It is useful in transfer learning problems where training and test domains are different. The LWE assigns weights to individual models based on their local behaviors at each test example, which allows it to adapt to different test examples.

The motivation behind this project is to improve the prediction accuracy in situations where the training and test domains differ, a common scenario in many real-world applications. This report provides an overview of the project, detailing the different steps in the algorithm, the experiments conducted and results.

The report is structured as follows: Section 2 surveys the relevant literature focussing on Sample Selection Bias and Covariate Shift , Section 3 describes the proposed project, Section 4 provides details on the dataset used, Section 5 outlines the experiment performed, Section 6 presents the results, Section 7 discusses future work, and Section 8 concludes the report.

2 Literature Survey

Traditional ensemble methods assign model weights based on the training set or fixed prior weights. However, these methods may not be suitable for transfer learning problems where the training and test domains follow different distributions. The following authors discuss the same issue.

Fan et al. (2006) [1]: This paper explores how data selection bias can affect building accurate models. In knowledge transfer, we are using a model trained on one dataset (source domain) to predict on another dataset (target domain). If the data collection processes for these datasets are different, it can lead to selection bias. The paper proposes using model averaging and unlabeled data from the target domain to reduce this bias and improve the transferred knowledge.

Huang et al. (2007) [2]: Similar to Fan et al., this paper also focuses on sample selection bias and how it impacts learning algorithms. In knowledge transfer, this bias can occur if the training data doesn't well-represent the situations we want to predict on. The paper introduces a method for correcting this bias by matching the distributions of the source and target data

in a special mathematical space. This can improve the transfer of knowledge by reducing the mismatch between the training and testing data.

Shimodaira (2000) [3]: This paper deals with a specific type of covariate shift where the factors influencing the data (covariates) differ between the source and target domains. In knowledge transfer, this can happen if the contexts or situations where the data was collected are very different. The paper proposes a technique for weighting the training data to account for this difference and improve the model's generalizability to the target domain.

Basically Fan et al. [1] and Huang et al. [2] talk about **Sample Selection Bias** which means that the training data doesn't represent the full range of possibilities you might encounter later. And Shimodaira [3] talks about **Covariate Shift** which means that the conditions or context where the data was collected are different between the training and testing situations.

3 Methods and Approaches

Clustering Manifold Assumption: The researchers assume that the dataset follows the clustering manifold assumption according to which for a sample x if the neighbouring points belong to class y then the sample x also belongs to class y.

The LWE uses a graph-based approach to approximate the optimal per example weight under the “clustering-manifold” assumption that P(x) is related to P(y|x). It constructs two graphs for each test example and a base model to be combined, and approximates the model weight as the similarity between the local structures around the test example in the two graphs. The LWE also includes a local structure-based adjustment mechanism to handle cases where the concepts carried by all the models conflict with the actual concept at the test example, meaning the average similarity value is lesser than a given threshold value.

This report discusses the Locally Weighted Ensemble (LWE) framework, a technique for combining multiple models in situations where the training and testing data come from different distributions (known as transfer learning). Here we'll focus on two key aspects of LWE: graph-based weight estimation and local structure-based adjustment. These steps play a crucial role in assigning appropriate weights to different models and potentially adapting predictions based on local information.

The Challenge: Conflicting Models and Uncertain Regions

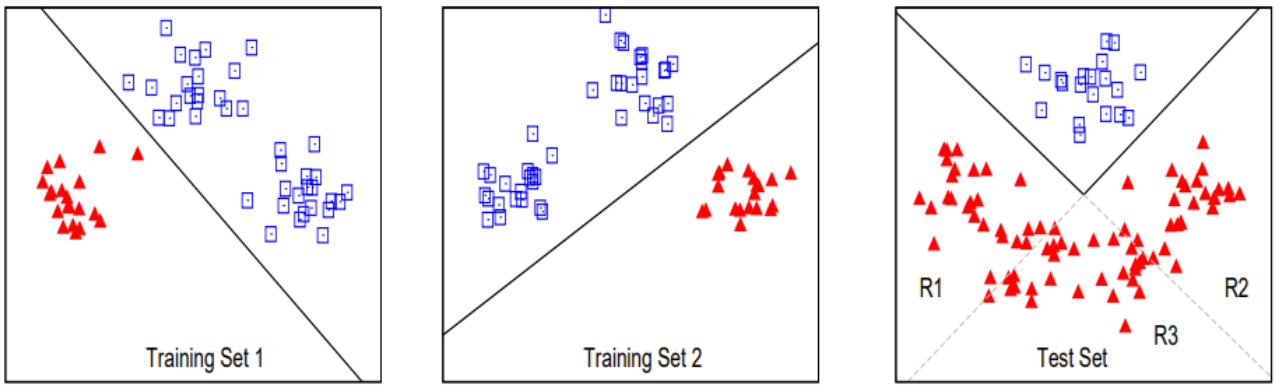

Figure 1: The figure shows two training sets with conflicting concepts and straight-line decision boundaries. The test set has a V-shape decision boundary.

This scenario involves two training datasets with conflicting decision boundaries and a test set with a V-shaped boundary. Simply combining the training data or the trained models (M1 and M2) would lead to inaccurate predictions in the uncertain regions (R1 and R2) of the test set.

The ideal solution would be a "locally weighted" ensemble framework that assigns higher weights to models based on their performance in specific regions of the test set. For example, M1 should have a higher weight in R1 and M2 in R2. This

approach leverages the strengths of each model in areas where they excel. The following sections introduce the Locally Weighted Ensemble (LWE) framework, which dynamically adjusts weights based on local model behavior.

Graph-Based Weight Estimation: Finding the Right Model for the Job

LWE addresses this challenge by assigning weights to each model based on its predicted performance for a specific data point (test example) in the new dataset. Here's how it works:

1. **Clustering the Test Set:** We start by grouping similar data points in the new dataset into clusters. This helps us understand the underlying structure of the data. The researchers have used a software called CLUTO for clustering, however I have simply used K means Clustering.
2. **Building Neighborhood Graphs:**

   In graph based weight estimation, we construct two graphs **GT** based on clusters made by the clustering algorithm, and **GM** based on the corresponding models predictions for each base model(that is SVM, LR and Decision Tree), for all the test samples. Then we compare the similarity between the two graphs. The similarity is normalised in order to calculate the weights that will be assigned to each base model(that is SVM, LR and Decision Tree).

3. **Comparing Local Structures:** The core idea is that if a model's predictions (connections in GM) align well with the underlying data structure (connections in GT) around a specific point (x), then the similarity value will be high hence, that model is likely to be reliable for predicting the class of x,
4. **Weight Calculation:** We calculate a similarity score (s(GM, GT; x)) that reflects how well the connections between models (GM) and clusters (GT) match around x. This score essentially tells us how much the model's predictions agree with the local structure of the data.
5. **Assigning Weights:** Finally, each model receives a weight (W_{Mi},x) proportional to its similarity score for a specific data point (x). Models with higher scores (better alignment with local structure) get higher weights, signifying their greater influence on the final prediction for x.

This approach gives more weight to models that consistently predict the same class for data points residing in similar regions (clusters) of the new dataset. This helps LWE leverage the strengths of each model in areas where their predictions are likely to be accurate.

Local Structure-Based Adjustment: When Models Disagree:

While weight estimation helps identify reliable models, there might still be situations where all models perform poorly for a specific data point (x). This could happen if the local structure (cluster) around x doesn't reflect the true class distribution. Here's how LWE handles such scenarios:

1. **Average Similarity Score:** LWE calculates an average similarity score (savg(x)) for all models around x. This score represents the overall agreement between model predictions and the local structure.
2. **Threshold Check:** If the average score (savg(x)) falls below a predefined threshold (δ), it suggests that none of the models are reliable for predicting the class of x. This might indicate a region where both models struggle due to limitations in their training data.
3. **Leveraging Local Structure:** In such cases, LWE discards the model predictions and relies solely on the local structure (cluster) of x in the new dataset. It predicts the class of x by looking at the majority class label of other data points within the same cluster (C) whose predictions from the weighted ensemble (E) were reliable (i.e., their savg(x) was above the threshold).

Local structure-based adjustment acts as a safety net. When models disagree, and the local structure suggests a potential mismatch between training and testing data.

3.1 Work Done

I have implemented LWE model in 2 steps::

Step1: pLWE (Partial Locally Weighted Ensemble) which is based only on Graph based weight estimation and does not consider Local Structure based adjustment.

Step2. Local Structure based adjustment

1. In this step we first set a threshold which we have set at 0.5, meaning at least 50% of the edges between the two graphs Gm and Gt should be similar. Then we calculate the average similarity for each test sample that is s_avg.

2. So if average similarity for a test sample is greater than 0.5 then the final prediction gotten from pLWE will be used for that test sample if not then we discard the prediction made by the pLWE model and consider a new unsupervised clustering model, (which in our case is K means Clustering) to make the predictions.

3. K means clustering has been used because the research paper mentions Unsupervised algorithm for clustering that is based on neighbors of x, although KNN would be a more appropriate choice for this but since it does not form clusters I decided to go with K means clustering algorithm.

4. Now all the test samples that have s_avg less than 0.5, are clustered using K means clustering and separated into 4 different test clusters. 4 test clusters because there are 4 sub categories in the test data.

5. Now after test clusters are formed reliable clusters are made that contain test samples for which the s_avg is greater than 0.35. Here I have taken 0.35 because all test samples have s_avg less than 0.4. So in order to make reliable predictions 0.35 was selected as an appropriate value.

6. Now we calculate the final probability for a test sample x belonging to say class 0, by dividing the no of reliable samples that also have class label 0, in that cluster to which x belongs, divided by the the total number of reliable test samples in that particular reliable cluster. All the reliable points are put in a cluster called c, called reliable cluster. For each cluster there is a corresponding reliable cluster. For example let us say test example x belongs to a cluster, then
probability that x belongs to class 0 = (no of reliable samples in cluster c with label 0/ total number of reliable samples in cluster c).

4 Data set Details

The following two diagrams represents how the subcategories are spread in a given space. It is clear that the training dataset can be easily clustered compared to the testing dataset. The four different colours represent four different subcategories. The text data is first converted into vectors using the TF-IDF Vectorizer followed by dimensionality reduction using PCA method so that data can be processed faster.

After which t-SNE is applied to the PCA-reduced data for both training and testing sets. (t-SNE is a tool to visualize high-dimensional data. It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. t-SNE has a cost function that is not convex, i.e. with different initializations we can get different results).

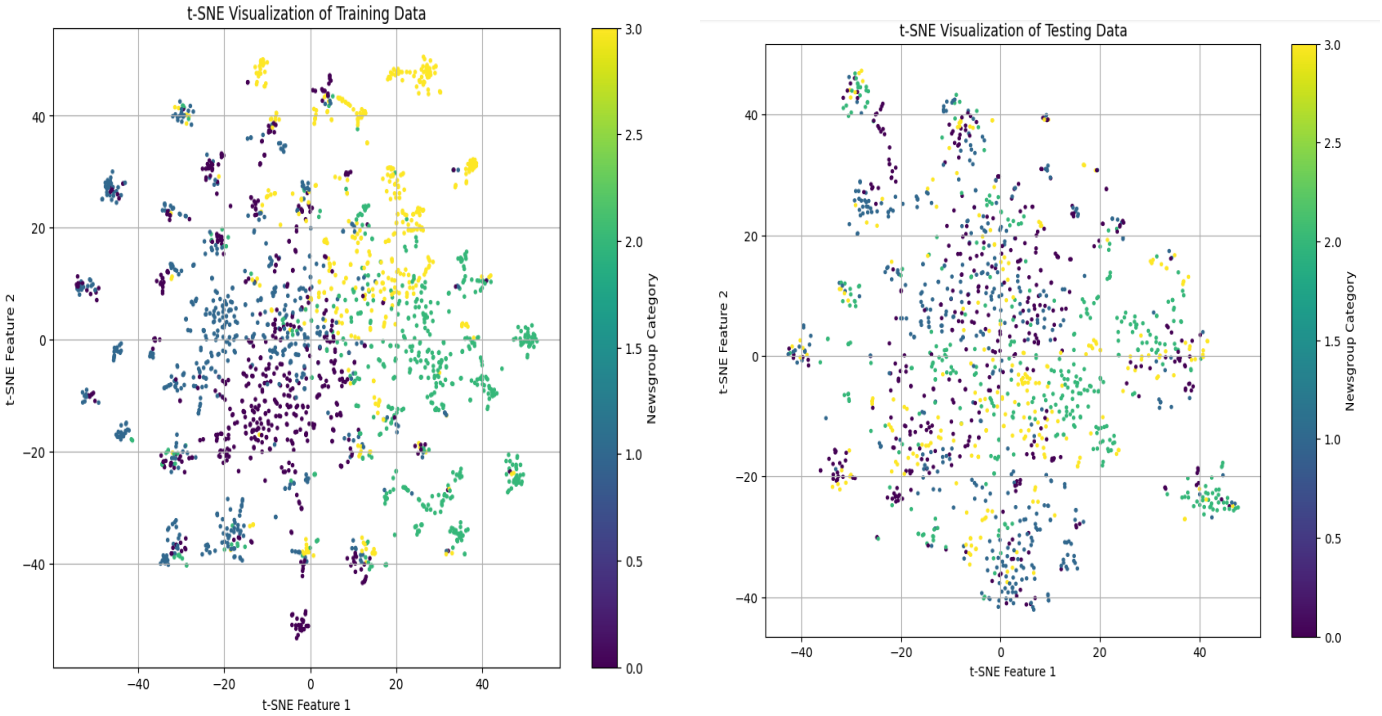

Figure 2: The two graphs represent the spread of the 4 different subcategories from Rec and Talk categories, which have been separated into training and test sets.

Data Set	$\mathcal{D}_i$	$\mathcal{D}_o$
comp vs sci	comp.graphics comp.os.ms-windows.misc sci.crypt sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.med sci.space
rec vs talk	rec.autos rec.motorcycles talk.politics.guns talk.politics.misc	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc
rec vs sci	rec.autos rec.sport.baseball sci.med sci.space	rec.motorcycles rec.sport.hockey sci.crypt sci.electronics
sci vs talk	sci.electronics sci.med talk.politics.misc talk.religion.misc	sci.crypt sci.space talk.politics.guns talk.politics.mideast
comp vs rec	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware rec.motorcycles rec.sport.hockey	comp.os.ms-windows.misc comp.windows.x rec.autos rec.sport.baseball
comp vs talk	comp.graphics comp.sys.mac.hardware comp.windows.x talk.politics.mideast talk.religion.misc	comp.os.ms-windows.misc comp.sys.ibm.pc.hardware talk.politics.guns talk.politics.misc

Figure 3: The table represents how different sub categories were divided into training and testing dataset.

(Source: Dai, W., Xue, G. R., Yang, Q., & Yu, Y. (2007, August). Co-clustering based classification for out-of-domain documents. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 210-219))

Di represents categories belonging to training dataset while **Do** represents categories belonging to the testing dataset.

The 20 Newsgroups dataset was used for the implementation of the LWE. The 20 newsgroup dataset is a popular dataset and can be imported from scikit using a command such as “from sklearn.datasets import fetch_20newsgroups”.

This dataset is a collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups. The TRAINING AND TESTING dataset had different subcategories an example of which is given below for an Rec vs Talk classification task.

training_set = ['talk.politics.guns', 'talk.politics.misc', 'rec.autos', 'rec.motorcycles']

testing_set = ['rec.sport.baseball', 'rec.sport.hockey', 'talk.politics.mideast', 'talk.religion.misc']

5 Experiments

The experiment aims to assess the effectiveness of both steps in LWE: weighted ensemble and local structure-based adjustment. LWE is compared against various ensemble methods such as SMA, pLWE and LS-SVM.

Experimental Setup:

- **Winnow** (WNN): This algorithm comes from the SNoW learning package. The code was unavailable for Winnow hence Decision Tree Classifier was used instead.
- **Logistic Regression** (LR)
- **Support Vector Machines** (SVM).

The above mentioned algorithms are used as base models for the LWE algorithm and their performance is also compared with that of the LWE MODEL. The Locally Weighted Ensemble (LWE) framework is compared with several other learning algorithms which are used to baseline the performance given as follows:

- **Simple Model Averaging (SMA):** This is a basic ensemble method where all models are combined with equal weights.

- **Partial Locally Weighted Ensemble (pLWE):** This method uses the weighted ensemble approach from LWE but skips the local structure-based adjustment step.

- **Least Square Support Vector Machines (LS-SVM):** This is a supervised learning which is being used to compare the performance of LWE. The researchers use Transductive Support Vector Machines (T-SVM), however due to unavailability of code LS-SVM was used.

Software: The researchers have used a clustering package CLUTO to group similar data points (clustering) in the test set, however I have decided to K means clustering instead.

6 Results

Expected Results:

Accuracy

Methods	20 Newsgroup					
	C vs S	R vs T	R vs S	S vs T	C vs R	C vs T
WNN	0.6554	0.5938	0.7942	0.7557	0.8926	0.9341
LR	0.7349	0.7217	0.7885	0.7904	0.8334	0.9176
SVM	0.7118	0.6824	0.7816	0.7577	0.8156	0.9389
SMA	0.7272	0.6845	0.7980	0.7806	0.8563	0.9348
TSVM	0.7697	0.8995	0.8996	0.8559	0.8964	0.8826
pLWE	0.7872	0.7217	0.8845	0.8330	0.9193	0.9664
LWE	**0.9744**	**0.9923**	**0.9823**	**0.9692**	**0.9816**	**0.9890**

Mean Squared Error

Methods	20 Newsgroup					
	C vs S	R vs T	R vs S	S vs T	C vs R	C vs T
WNN	0.2775	0.2968	0.1575	0.1978	0.0851	0.0525
LR	0.2057	0.2036	0.1567	0.1624	0.1340	0.0613
SVM	0.2140	0.2353	0.1644	0.1826	0.1360	0.0453
SMA	0.2030	0.2183	0.1349	0.1614	0.0979	0.0430
TSVM	0.1749	**0.1080**	0.1128	0.1281	0.1198	0.1061
pLWE	0.1795	0.2027	0.1029	0.1399	0.0699	0.0302
LWE	**0.0965**	0.1409	**0.0384**	**0.0534**	**0.0308**	**0.0140**

Figure 4: The above figure shows the expected results for 20 newsgroup dataset corresponding different binary classification tasks.

Received Results:

	Accuracy						Mean Squared Error					
	C vs S	R vs T	R vs S	S vs T	C vs R	C vs T	C vs S	R vs T	R vs S	S vs T	C vs R	C vs T
Decision Tree	0.589	0.640	0.651	0.616	0.706	0.725	0.411	0.360	0.349	0.384	0.294	0.275
Logistic Reg.	0.673	0.623	0.728	0.720	0.829	0.892	0.327	0.377	0.272	0.280	0.171	0.108
SVM	0.736	0.659	0.736	0.731	0.848	0.921	0.264	0.341	0.264	0.269	0.152	0.079
SMA	0.694	0.6338	0.7332	0.7156	0.833	0.8931	0.305	0.3661	0.2667	0.2843	0.166	0.1068
LS-SVM	0.7232	0.2895	0.7780	0.7803	0.821	0.8828	0.2767	0.7104	0.2219	0.2196	0.178	0.1171
p-LWE	0.6381	0.6296	0.6967	0.7078	0.7996	0.8541	0.3618	0.3703	0.3032	0.2921	0.2003	0.1458
LWE	0.5973	0.7814	0.5025	0.4843	0.8268	0.8342	0.4026	0.2185	0.4974	0.5156	0.1731	0.1657

Figure 5: The above figure shows the expected results for 20 newsgroup dataset corresponding different binary classification tasks.

The results show that the LWE algorithm performed relatively well on the R vs T, C vs R, C vs T classification tasks giving higher accuracy and lower MSE compared to other algorithms. Even though the accuracy and MSE are not as good as compared to the Expected results mentioned in research paper, however the received results prove the point that the LWE algorithm has better performance compared to other mentioned algorithms.

Reasons for Variations in Results:

1. For C vs S, R vs S and S vs T classification tasks the LWE accuracy and MSE is not good compared to the expected results. I t must be noticed that the dataset belonging to S category is common in all three cases, and the lower performance in the results could be attributed to the discrepancy in the data, however I could not find any concrete proof for this claim.
2. Another reason could be because of the small change in the algorithm, as in the second step of the LWE algorithm, instead of using the prediction for datapoints (with s_avg > threshold) from pLWE step, I have calculated the predictions for all the datapoints by taking two different thresholds, that is threshold= 0.5 for comparing the similarity in order to decide whether Structure based adjustment step is needed or not and threshold=0.35 or 0.34 in order to form clusters with reliable datapoints.
3. Also as mentioned earlier the researcher have used a clustering package called CLUTO, whereas I have K means clustering because of the unavailability of CLUTO which could also lead to differences in the results.

7 Future Work

- Future work will focus on exploring other datasets using the LWE algorithm and testing the performance on that dataset.
- Trying to replicate the exact algorithm and testing if the performance improves, by removing the small modification in the Local Structure based adjustment step (as described in **Reasons for Variations in Results** part 2).

8 Conclusion

- LWE offers an advantage by assigning weights based on local behaviour (using clusters), it leverages the strengths of each model instead of blindly combining predictions that might be inaccurate in certain regions. Furthermore, the local structure-based adjustment provides a backup mechanism when models fail to generalize well using the Local Structure based adjustment step.
- The LWE algorithm is effective for transfer learning problems especially when the training data and testing data are not necessarily similar. It effectively combines multiple models by assigning weights based on their local behaviors at each test example.
- The implementation of the LWE on the 20 Newsgroups dataset demonstrated its effectiveness and potential for future applications as consistently gave average accuracy around 70% and an average MSE around 0.25. There is scope for improvement in the present version od the algorithm and step discussed in future could be implemented to improve the performance.

9 References

1. W. Fan and I. Davidson. On sample selection bias and its efficient correction via model averaging and unlabeled examples. In Proc. of SDM'07,2007, https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.29
2. J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Sch¨olkopf. Correcting sample selection bias by unlabeled data. In Proc. of NIPS' 06, pages 601–608. 2007, https://proceedings.neurips.cc/paper/2006/hash/a2186aa7c086b46ad4e8bf81e2a3a19b-Abstract.html
3. H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference, 90(2):227–244, 2000, https://www.sciencedirect.com/science/article/pii/S0378375800001154
4. Saigal, P., & Khanna, V. Multi-category news classification using Support Vector Machine based classifiers. *SN Applied Sciences*, *2*(3), 458., (2020), https://link.springer.com/article/10.1007/s42452-020-2266-6
5. Dai, W., Xue, G. R., Yang, Q., & Yu, Y. (2007, August). Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 210-219).,2007, https://dl.acm.org/doi/abs/10.1145/1281192.1281218

6. Ieracitano, C., Adeel, A., Gogate, M., Dashtipour, K., Morabito, F. C., Larijani, H., ... & Hussain, A. (2018). Statistical analysis driven optimized deep learning system for intrusion detection. In *Advances in Brain Inspired Cognitive Systems: 9th International Conference, BICS 2018, Xi'an, China, July 7-8, 2018, Proceedings 9* (pp. 759-769). Springer International Publishing.,2018, https://researchonline.gcu.ac.uk/ws/portalfiles/portal/26504106/BICS_2018_paper_65.pdf
7. Danny Vanpoucke, https://github.com/DannyVanpoucke/LSSVMlib/blob/master/LSSVMlib/LSSVMRegression.py, Accessed on 20th March, 2024.